SOFTWARE MAINTENANCE: RESEARCH AND PRACTICE, VOL. 9, 85–101 (1997)

**Practice**

# Model-based Assessments Applied to Software Maintenance Processes

DERICK E. MASENGALE[1*] AND MICHAEL J. SCHRANK[2]

[1]*EDS Corporation, Government Services Division, Skyline 2 Suite 1200, 5203 Leesburg Pike, Falls Church, Virginia 22041, U.S.A.*

[2]*The MITRE Corporation, 1820 Dolley Madison Blvd., McLean, VA 22102, U.S.A.*

**SUMMARY**

**The roles and responsibilities of software maintenance organizations are often misunderstood by individuals from both inside and outside the organization. This paper presents a practical and uncomplicated assessment methodology that can be efficiently applied to assess any organization in order to identify and understand its current software maintenance process and accompanying roles and responsibilities. This understanding combined with practical experiences drawn from the software maintenance community can aid in identifying meaningful software process improvements. The paper discusses observations and conclusions that resulted from our experiences with this model-based method of process assessment. ©1997 by John Wiley & Sons, Ltd. J. Software Maintenance 9: 85–101, 1997.**

(No. of Figures: 6. No. of Tables: 2. No. of Refs: 7.)

KEY WORDS: software maintenance; process modelling; process improvement; process assessment; personnel roles; personnel responsibilities

## 1. INTRODUCTION

### 1.1. Evolution of process assessments

For several years MITRE has been supporting an ongoing effort to assess the current and future capabilities of military software maintenance organizations (SMOs). This is achieved through the performance of the MITRE-developed and refined baseline assessment process. The objectives of a baseline assessment are to determine the current and future software maintenance capabilities at a specific organization; evaluate the software products being maintained and processes used in performing the maintenance; review the organizational infrastructure; identify areas of technical, programmatic and cost risk; estimate the costs associated with maintenance; and identify actions which could reasonably be taken

* Correspondence to: Derick E. Masengale, EDS Corporation, Government Services Division, Skyline 2 Suite 1200, 5203 Leesburg Pike, Falls Church, VA 22041, USA. Email: dmasenga@med.osd.mil

CCC 1040–550X/97/020085–17$17.50

*Received 1 June 1996*
*Revised 18 September 1996*

to improve business efficiency and software quality. These assessments have been performed to successfully address these and many other issues and questions.

Baseline assessments differ from other process assessments, such as the Software Engineering Institute's (SEI) assessments (e.g., software capability evaluation (SCE), software process assessment (SPA), etc.), in that they consider the entire organization and other relevant areas that influence the maintenance of software in addition to the processes. These areas include past and current products, laboratories, support software, testing facilities, software distribution and installation, training/product support activities, and resources, etc. The baseline assessment also includes current cost analysis and future cost estimation as part of the effort. This methodology provides a complete assessment of the overall 'health' of an SMO.

The baseline assessment methodology has its roots in previous forms of assessments. The best fundamental concepts were extracted from earlier process assessment and improvement initiatives such as total quality management (TQM), SEI's capability maturity model (CMM), SCE, SPA, ISO 9000, and others. TQM applies quantitative methods and resources to improve, among others, all the processes within an organization. The CMM applies process management and quality improvement concepts to software development and maintenance. The SCE, using the CMM, formalizes the assessment of a software organization by an external audit team with the results being used by acquisition agencies. The SPA (also based on the CMM) provides a software organization with the ability to perform an internal self-assessment on their software processes and use the results within that organization. The SPAs have since become known as CMM-based appraisal for internal process improvement (CBA IPI). ISO 9000 series quality standards (including software) provide a basis for defining and contractually specifying minimal quality assurance and management requirements.

## 2. BASELINE ASSESSMENTS

### 2.1. Overview

Baseline assessments have been successfully performed on organizations that maintain management information systems (MIS), and those that maintain embedded software on a military platform (Schrank *et al*., 1995). During the performance of a baseline assessment, subteams are formed to address system, software and cost issues. The system subteam focus includes: program status, system status, program management, tasking and funding, developmental and operational testing, and facilities. The software subteam focus includes: software engineering processes and procedures, metrics, deliverable and support software products (code and documentation), configuration management (CM) and quality assurance (QA). The cost subteam focus includes: data collection for cost analysis, budgeting and tracking, accounting practices, and determination of funding requirements.

Baseline assessments generally follow a four phase approach: (1) planning and preparation, (2) the site visit, (3) reporting and (4) follow-up. The interaction of these four phases is depicted in Figure 1.

During the planning and preparation phase, the assessment team gains an understanding

 

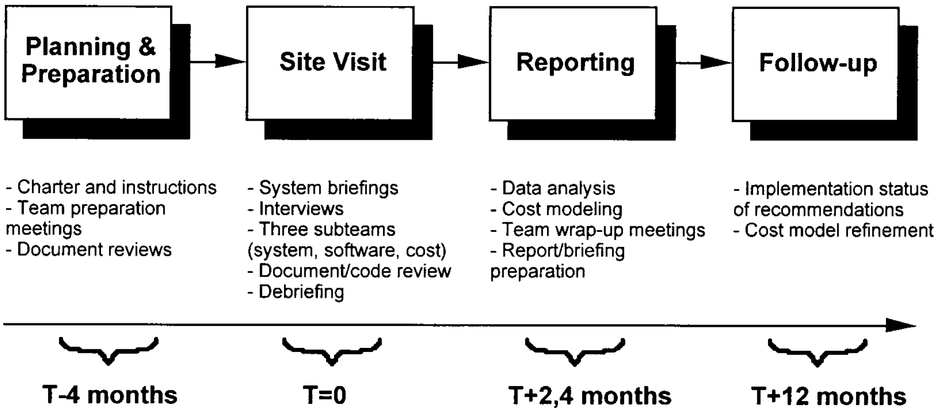


*Figure 1. Baseline assessment process*

of the project. Based on the scope of the assessment, the team determines the specific issues that need to be addressed, the persons to be interviewed during the site visit, and the artefacts to be examined. The site visit phase provides the detailed information required to assess thoroughly the capabilities of the organization. The reporting phase consists of the exit debriefing to the project management, the 'should cost' economic analysis, issue clarification and the preparation of a written report. Follow-up consists of monitoring and/or reassessment to evaluate the implementation status of the recommendations. Each of the four phases of the assessment is described in greater detail below.

## 2.2. Planning and preparation

A baseline assessment is carefully planned to examine all of the program management, system engineering and integration, software engineering, validation and verification, CM and QA processes, capabilities and products. Independent validation and verification, if present, should also be examined.

Initially, team members should understand the history and objectives of the SMO as well as its planned schedule and products. They should examine the available management documents (life cycle management plans, test plans, software development/maintenance plans, CM plans and QA plans). If available, they should also review the software standards and procedures manual and detailed procedures to be applied to the software maintenance process.

## 2.3. Site visit

The purpose of the site visit is to collect the data necessary to assess the SMO capabilities and processes and to collect cost data in support of cost estimation. Also, the site visit allows a determination of whether the SMO conforms to their published plans, standards and procedures. Entrance briefings are conducted by the assessment team to describe the focus of the assessment process and schedules, and by the SMO to provide

 

technical and programmatic information about the system. A tour of the facilities usually occurs followed by interviews in each of the three subteam areas of systems, software and cost. Since each subteam has a slightly different focus the interviews are conducted separately. An exit interview provides an opportunity to clear up misunderstandings and allows project staff to present any information that they feel the team failed to consider.

## 2.4. Reporting

The team prepares a written final report, incorporating any additional information gathered in the exit interview. The report is organized to highlight the most significant findings and recommendations, and to present the results of the cost estimation. The objective of the report is to present a clear picture of the status of the SMO, to provide recommendations for problems that have been found, and to provide a cost estimate for the organization.

## 2.5. Follow-up

After the report is complete, actions to resolve deficiencies and implement recommendations provided in the report are undertaken by the sponsoring organization. Problems that are feasible and reasonable to correct should be converted to action items and assigned to appropriate individuals. A rationale should be provided for those items that are not to be corrected. The changes should be tracked to ensure they occur and are effective and the closure of action items should be documented. In many cases, the best way to determine if the problems have been solved is through a follow-up assessment. This follow-up assessment is done after some time has passed (usually about a year) to allow for improvements to be implemented.

## 2.6. Evaluation of the baseline assessment process

During a baseline assessment, we are often asked by the SMO to recommend process improvement activities that can be incorporated into the current processes or even process changes that could be beneficial. Since the baseline assessment does not directly address process improvement (nor does an SPA or SCE), we investigated avenues for modifying the baseline assessment methodology for this purpose.

Process improvements have been studied and applied in many areas of industry for years. The most notable of which is the manufacturing industry. However, process improvement concepts have only been applied over the past decade to the software community, both commercial and military. Applying the manufacturing process improvement paradigm to the software maintenance process has proven very successful. The basic steps in process improvement are shown in Figure 2 (Humphrey, 1989).

While the software maintenance industry has markedly improved its ability to discover what activities within the software maintenance process require improvement, it is still struggling with how to implement effectively the improvements. As we evaluated our baseline assessment methodology we recognized that like the SPA and SCE, the baseline assessment process only really allowed us to address step 2 (identify and analyse areas

 

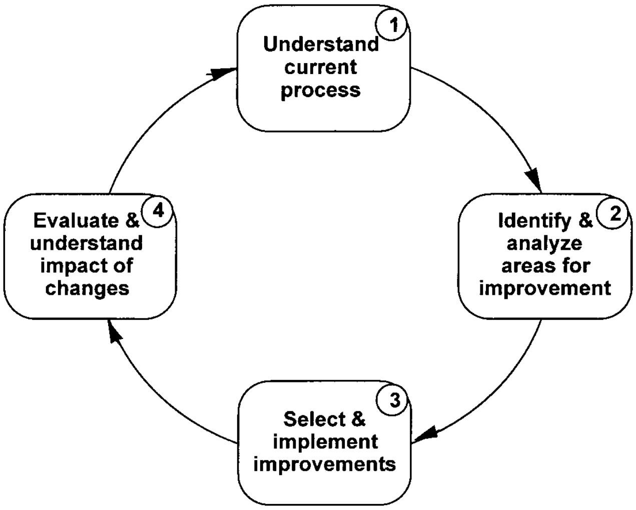


*Figure 2. Iterative steps in process improvement*

for improvement) of the process improvement model. We experienced difficulty in efficiently organizing and analysing the reams of data collected during an assessment. Finally, it was clear that to implement effectively the process improvements identified, we needed to clearly understand the current process, select and implement the improvements based on steps 1 and 2 of the process improvement model, and evaluate and understand the impact of changes made to the process. In an effort to address these needs, we incorporated process modelling into our baseline assessment methodology.

## 3. BASELINE ASSESSMENTS USING PROCESS MODELS

### 3.1. Three capabilities

Modelling is conducted to answer questions about the process being modelled. It is generally agreed that software process models must possess capabilities in the following three major areas:

1. representation power,
2. comprehensive analysis capabilities, and
3. forecasting capabilities.

Individuals within MITRE have experimented with incorporating process modelling into software process assessments. Based on the experience data collected by McGowan and Bohner (1993), using structured analysis and design technique (SADT) and $IDEF_0$ (Integrated DEFinition 0, a standardized notational subset of SADT) in process assessments, we selected $IDEF_0$ as our process modelling technique (Marca and McGowan, 1993).

## 3.2. Process modelling approach

### *3.2.1. Six activities*

McGowan and Bohner's model based process assessment (MBPA) approach uses thorough on-site interaction with the development organization. Conversely, a baseline assessment only allows for three to five days of on-site interaction with the SMO. As a result, meticulous preparation by the baseline assessment team is essential to ensure a successful assessment of the SMO. The process model and the time spent developing the model proved invaluable in our preparation effort. The model was developed and incorporated into the assessment process by performing the following activities:

1. collect and review all relevant SMO process documentation,
2. develop baseline process model using SMO documentation,
3. plan for SMO site visit,
4. conduct SMO site visit,
5. analyze information collected, and
6. develop a process improvement strategy.

Figure 3 demonstrates how this process modelling added to the original baseline assessment process. The items shown below the dotted line are the additional activities that are performed as part of the process modelling. The following discussion focuses primarily on how process modelling was incorporated into the baseline assessment process and then provides a summary of our experience and observations using this technique.

### *3.2.2. Collect and review all relevant SMO process documentation*

In order to build the model, one must collect as much documentation describing the SMO's software maintenance process as possible. Table 1 contains a partial list of documents one may find useful in developing a model.

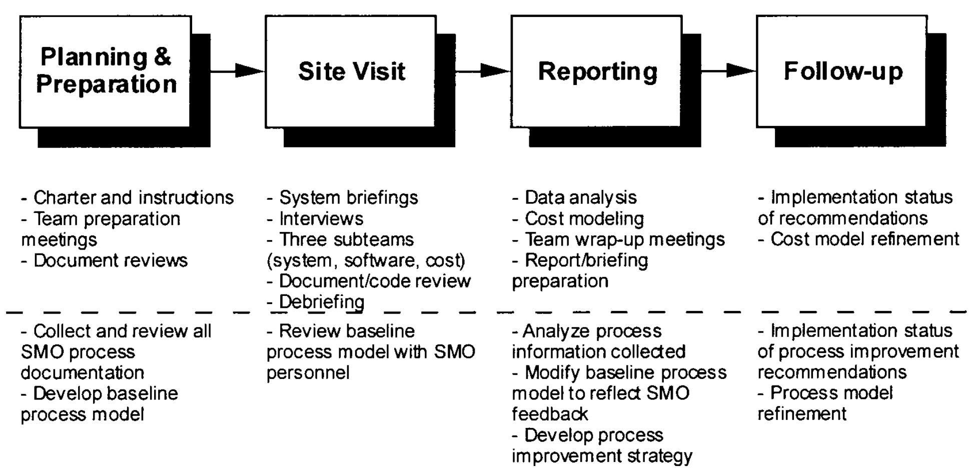


*Figure 3. Baseline assessment process with process modelling incorporated*

 

Table 1. Documents used to develop process model

| |
|---|
| Program-level management plan |
| System description documentation |
| Software development plan (SDP) |
| Software configuration management plan (SCMP) |
| Desktop-level configuration management (CM) procedures |
| Software configuration control board (SCCB) charter |
| Software quality assurance plan (SQAP) |
| Desktop-level quality assurance (QA) procedures |
| Software test plan (STP) |
| Computer resources life cycle management plan (CRLCMP) |
| Programmers handbook |

Indeed, if one is working with an organization possessing an SEI capability maturity model (CMM) level 1 rating, many of these documents may not formally exist. However, available documents often reference other organizational unique documents or informal procedures that describe in more detail a given activity or group of software maintenance activities. Unfortunately, it is almost certain that the SMO's existing documents contain inconsistencies and inaccuracies in their description of software maintenance activities. It is important to document any inconsistencies found, because these will be used during the final preparation for the site visit.

Typically an SMO performing government contract work will have produced the high level management plans (SDP, SCM, SQAP and CRLCMP) identified in the statement of work (SOW). It is usually the procedural level documents that are not produced by the SMO. What the modeller must do is create the model based on what is documented in the plans. This most likely will represent an 'idealized' picture of the SMO's operations. This 'idealized' model will be modified to reflect reality during the site interview.

### *3.2.3. Develop baseline process model(s) using SMO documentation*

Next, as summarized in Figure 4, the available documentation is used to develop the software maintenance process model(s). To create a model that most effectively supports process improvement, it is important to integrate all SMO processes and procedures into the maintenance model. As previously mentioned, the documentation for an SEI CMM level 1 process typically contains inconsistent, incomplete and inaccurate descriptions of process activities. Low quality documents can make the task of modelling the process more difficult and require the modeller to become 'creative' when describing poorly documented process activities (Bollinger and McGowan, 1991). This should not discourage the modeller; this activity provides great insight into an assessment of the quality of the SMO's documentation. Process activities that are poorly documented are most likely activities that are poorly understood and inefficiently performed by the SMO. These activities should be identified, documented, and explored in greater detail during the final preparation for the site visit.

Because many of a CMM level 1 organization's procedures and processes are not repeatable (they may be executed differently each time they are performed), ambiguous

 

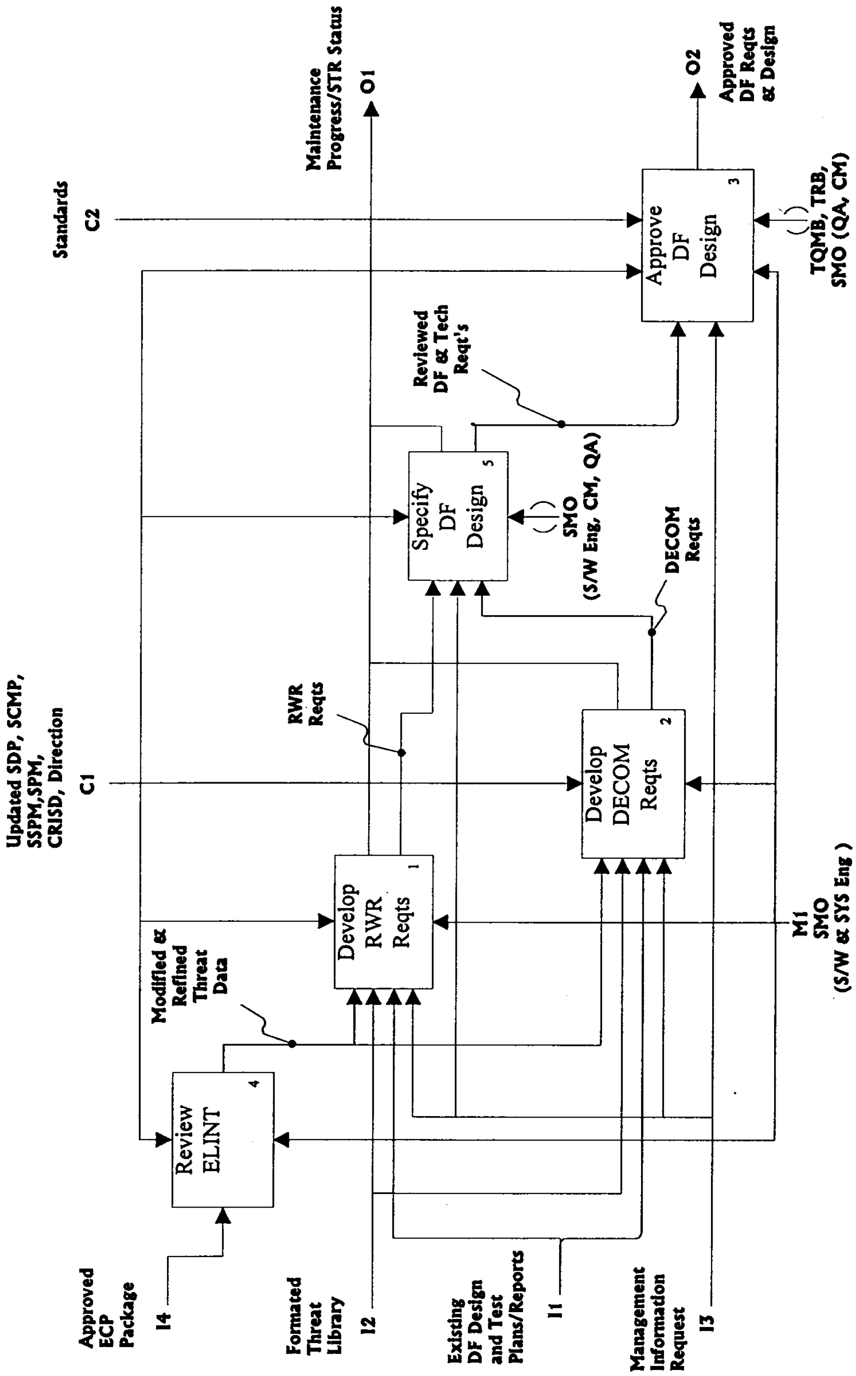


*Figure 4. Baseline for process model*

and conflicting descriptions of activities most likely exist. When this situation is identified, the following question will have to be answered by the modeller. How should an activity be modelled when two conflicting descriptions exist for the same activity? In such instances, the modeller must make decisions on how he/she believes the activity is most likely performed. It is not as important for the modeller to select the ‘right’ way as it is for the modeller to link the activities represented in the model to the document(s) that provide the descriptions. This highlights a very important point; any modelling tool that is selected should provide the capability to associate informational notes with each activity in the model. Capturing this information and having the cross-reference data available during review of the model with the SMO is crucial.

The modeller must also decide what level of detail the model is to depict. The level of detail most likely will be determined by the quality of the SMO’s documentation. It is likely that certain activities in the process will be described in great detail while other activities are described at a much higher level in more general terms. If possible, try to develop all activities in the model to the same level of abstraction. This will prove beneficial if the model is used to generate requirements management (RM), CM, QA and metrics plans/procedures.

One individual from the assessment team is given the responsibility of developing the model. This is to provide consistency and coherence in model construction. Even so, the entire team is responsible for reviewing the documentation and providing inputs to each iteration of the model during development. Because each team member may look at the process from a different viewpoint than the modeller, this activity usually produces a number of process questions and inconsistencies that have not been identified by the modeller. These questions and inconsistencies are documented and reviewed during the final preparation for the site visit. When the final revision of the baseline model is complete, the team then meets to plan for the site visit.

In summary, the goal of this phase is to create a model that depicts, in their proper context, all activities described in the SMO’s documentation. For a CMM level 1 organization, which by its very nature assumes that these activities are not well understood, this may prove elusive. Even if an organization has a model that appears to completely describe the maintenance process it is worth the time and effort to compare the model with the organization’s documentation. Experience shows that even if a process is documented, it is often not performed as per the documentation. Using the model during the interview process will allow the assessment team to quickly determine how each activity is truly performed. In contrast, while an SPA and SCE can determine if a particular CMM key practice area (KPA) is being performed it rarely determines how it is being performed. The ‘how’ is very important when determining ‘how’ to implement the corresponding process improvement.

### *3.2.4. Plan for the SMO site visit*

During this step, the baseline process model is compared with the KPAs (Paulk *et al*., 1991) and other industry best practices to identify areas of possible process weaknesses. The team will compare the model with CMM level 2 processes to obtain an initial feel for the quality of all major activities in the maintenance process. A list of questions will

be generated from this review and also from all the notes and questions documented during the development of the baseline process model. These questions may be asked of the appropriate SMO personnel during the interview process to provide a more in-depth understanding of current process activities.

### *3.2.5. Conduct SMO site visit*

At the site, interviews are conducted with management and engineers from the maintenance, CM, QA and testing organizations. During each interview, the accuracy of the model is discussed and the appropriate questions are asked from the prepared list. This is where the power of process models is most apparent. The nature of the model allows one to ask very focused questions and obtain a great deal of information in a very short period of time. Process models also are a mechanism for obtaining consensus on the activities of a process. We found it very useful to point to an activity in the model (for example, activity A1 in Figure 5) and then pose the following questions.

1. Is this activity actually performed at this point in the process?
2. Are the inputs correct and complete?
3. Are the outputs correct and complete?
4. Are the resources (i.e., personnel and facilities) required correct and complete?
5. Are the constraints for the activity correct and complete?

This information is then used to generate a model that more truly reflects how business is currently conducted within the organization. The SMO is provided with the revised model to review and provide final inputs. A final version of the model is developed using these inputs (reference Figure 6).

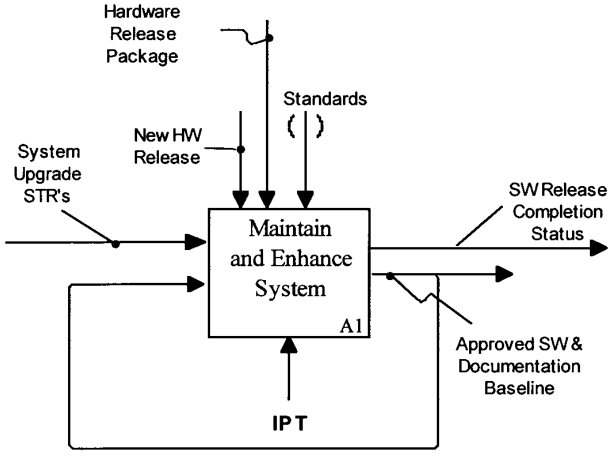


*Figure 5. Example of an IDEF$_0$ process model activity*

 

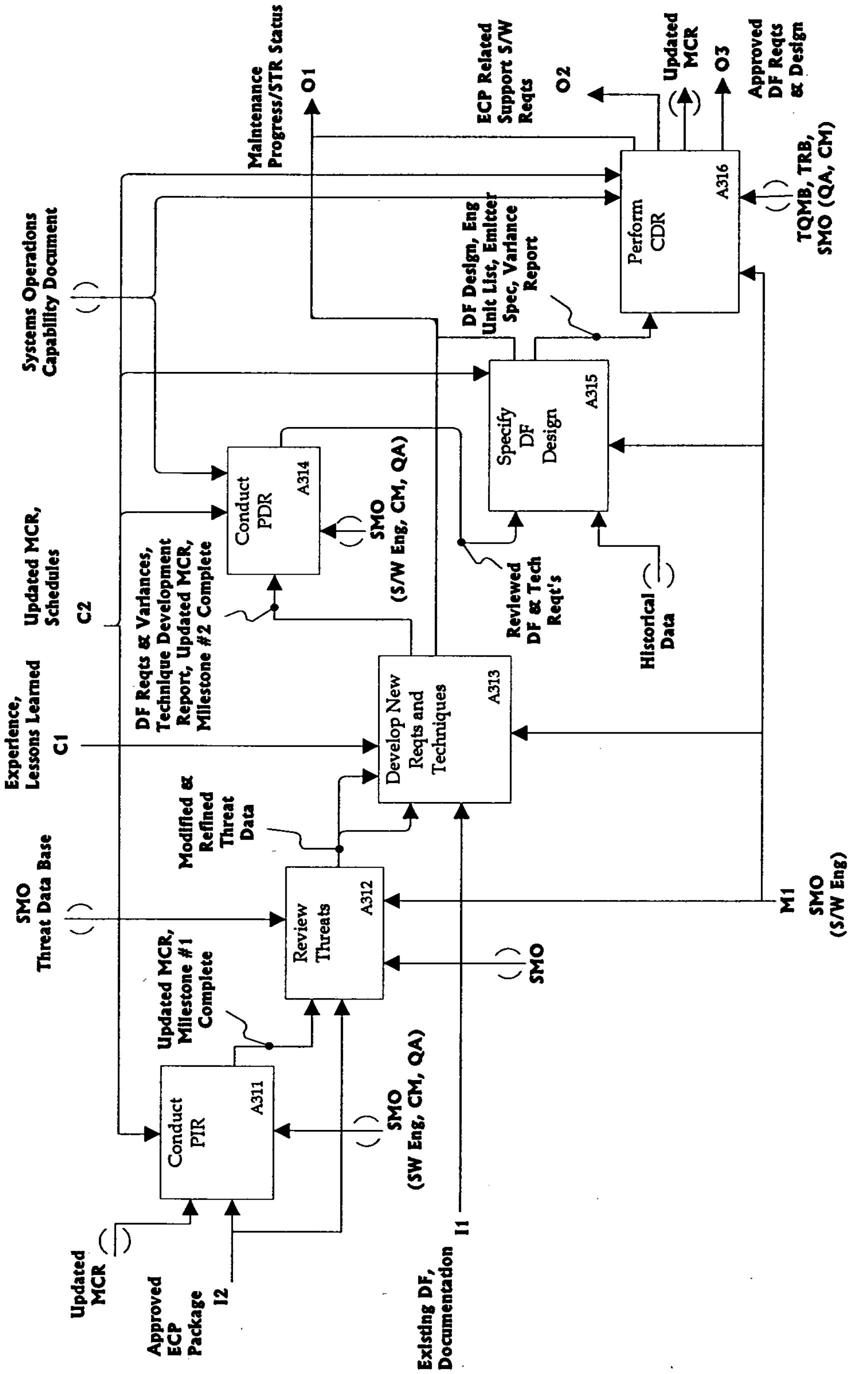


*Figure 6. DF design for revised model*

### *3.2.6. Develop a process improvement strategy*

After a final model is developed, the next step is to develop a process improvement strategy. The primary focus of our process improvement strategy is to increase the productivity of the maintenance staff while increasing the quality of all deliverables. It is not to ensure that the organization is SEI CMM compliant. For example, because these are maintenance organizations, they face issues that are not addressed in the CMM (e.g., replication, distribution, help desk activities). As a result, an organization could be fully compliant with the SEI CMM and still not be efficient and effective delivering products to customers. The following activities are all key components in developing an effective process improvement strategy.

- Compare the current process model with the KPAs defined in the SEI capability maturity model and document areas that require improvement.
- During interviews have SMO personnel point out areas of high frustration and perceived significant work-flow bottlenecks within the current maintenance process.
- Identify activities described in the baseline model that do not reflect current 'best' practice. Identify the documents that incorrectly describe the activities.

At this point, steps 1 and 2 of the process improvement paradigm (Figure 2) are complete and a process model exists to aid with step 4. The process of developing the initial model and reviewing the results with the SMO personnel provides all those involved with a clear understanding of the current process. Once the initial model has been reviewed by the SMO and modified accordingly, the quality of the CM, QA and project management (PM) solutions may be quickly assessed. It is highly recommended that the activities requiring improvement identified by the front-line maintainers be considered as highest priority when developing an implementation plan (step 3). After the activities or processes requiring improvement have been identified and prioritized, the model is then used to determine how to best implement the changes. Because the model pictorially describes how a process or activity is currently performed, the relationship between organizations and activities within the entire maintenance process is clearly depicted. The understanding of these relationships is crucial to effect process improvement. Furthermore, because the documents that describe the performance of an activity are identified on the model, it is easy to determine where the changes should be documented.

In many ways the ultimate success of any process improvement initiative is dependent upon the support of the front-line maintainers along with management's backing. The sooner these individuals see 'personal benefit' from the process improvement initiative the more likely they will become active participants in the program. Because the maintainers are required to implement, act upon, and collect data related to new procedures, their degree of compliance is crucial to the overall success of the program.

 

## 4. PRACTICAL EXPERIENCES

### 4.1. Success of model

This section describes our actual experience performing baseline assessments that included process modelling. We believe the incorporation of the process models into the Baseline Assessment proved to be a tremendous success. Not only did the process models fulfil our intended goal of organizing large amounts of data collected during an assessment, but the models also provided many unexpected benefits. We will continue to use and refine the incorporation of models into our baseline assessment process. Furthermore, it is our recommendation that any SMO with a CMM level 1 or 2 process should begin developing process models in a similar fashion as early in their process improvement program as possible.

One question we are nearly always asked is, can you quantify your results? Can you show a cost benefit? Without realizing it those inquiring are asking an unanswerable question. We certainly can, and do, track the cost of performing a baseline assessment, but a CMM level 1 organization *does not know what it costs to perform any activity within it's maintenance process*. Without this information it is impossible to answer the cost benefit question. However, when we are complete with the assessment we provide a model to the SMO that describes their current process and supports an activity-based costing approach. This model can then be used to determine the cost of performing the current activities and quantify the effectiveness of future process improvement initiatives (step 4 in Figure 2).

### 4.2. Effort required

The baseline assessment is structured to ensure minimal impact on the maintenance organization being assessed. As a result, preparation before the site visit is key to the success of the assessment. We found that it took one individual three months of full-time work to learn SADT, $IDEF_0$; review and study documentation received from the SMO; and then develop a fairly detailed model. The model contained enough detail so that desktop-level CM and QA procedures could be developed from it. Based on our more recent experience, if someone was versed in $IDEF_0$ it would probably take about a month to develop a model, depending on the detail and quality of documentation received. The remainder of the team (five individuals) worked anywhere form 25 to 50% of their time during this period reviewing SMO documentation and model iterations. One week was spent at the SMO site, followed by about another four staff months of work to assemble the data, formulate recommendations and generate the report.

It should be noted that several of the assessments also included an examination of the facilities and an economic analysis of the SMO. In baseline assessments where these activities were not included (termed mini-assessments), the manpower required to perform the assessment was reduced by nearly 50% (reference Table 2). This means that an initial assessment of an SMOs software maintenance process would require about ten staff months of total effort over six to eight calendar months. Therefore, an entire baseline assessment may be performed for approximately 18 staff months and be completed in

 

Table 2. Assessment comparisons

| Assessment phase | Full assessment | Mini-assessment |
|---|---|---|
| Planning and preparation | 9·5 staff months | 6·0 staff months |
| Site visit | 1·5 staff months | 1·0 staff months |
| Reporting | 7·0 staff months | 3·0 staff months |
| Follow-up | N/A | N/A |

less than five calendar months. This is a tremendously important feature of a baseline assessment.

In contrast to an SCE, at completion of the model-based assessment, a detailed model of the current maintenance process would exist, and areas identified as needing improvement and initial recommendations for implementing the process improvements would be available. Because each activity depicted in the model identifies the document(s) that describe how the activity is performed, this serves as a great aid in implementing changes in the process.

## 4.3. Summary of case study results

The results of incorporating process models into the baseline assessment methodology far exceeded our expectations. The process model proved to be more than just a tool to organize the information collected about the SMO. In contrast to the questionnaire methodology (employed in an SEI SPA or SCE), the model allowed us to see each piece of data within the context of the entire software maintenance process and then analyse the interrelationships between the data collected before and during the assessment. This ability to see the interrelationships between activities within the entire software maintenance process is essential to successfully implementing change into the process. Without this insight, well intended changes can actually make matters worse.

The following experience was noted during the assessment of an SMO. The SMO being assessed had recognized the need to improve its configuration management process. Management directed a group of individuals from the QA and CM organizations to modify the current CM process to make it more formal. The QA and CM organizations looked for industry best CM practices to incorporate into the process (a good idea). After completing the research they developed a new CM process based on the waterfall model for software development. When we performed the site visit, the organization was preparing to implement the new procedures. After developing the final process model with the SMO, we found they were not developing the software using a waterfall model but using an iterative approach instead. The SMO would identify possible requirements, modify the software, and then test the software to see if the requirements were correct. If problems were found during testing, the problem could be either a software error or an incorrect requirement. If it was determined to be an incorrect requirement, the whole process would be repeated.

For this process to work efficiently, the requirements documentation is not placed under

 

configuration control until after unit testing is complete. Because the waterfall model requires that the requirements be under CM control before development begins, it was clear that the new CM process would not work for the organization—it would have to be rewritten. Without the model the new process would have been incorporated into the maintenance process and frustration between all affected organizations would have been the result—not a better CM process. If the model had been developed earlier in their process improvement program, the money spent to rewrite the CM procedures could have been saved.

As previously mentioned, a number of unexpected benefits were observed during the use of the models in the assessment process. For example, the model proved to be a very efficient method of identifying and documenting the current software maintenance process. While this may seem to be a trivial activity, our experience shows that an SMO with a CMM level 1 process usually does not understand its current software maintenance process. A common understanding of the software maintenance process across all affected teams within the organization can be important in obtaining process improvement buy-in. Using the process model as a tool for understanding allows multiple organizations to see the constraints and information requirements of all organizations involved with a particular activity.

The model can also aid in determining which documents should be produced to describe the software maintenance process and associated procedures. When multiple procedures describe the same activity, one set of procedures can be selected to describe the activity, thus minimizing the documentation required to describe the software maintenance process and clarifying which organization 'owns' a given activity.

We found the model to be a great document configuration management aid. Because the model contains references to every document that describes an activity, when an activity is modified the model can be used to identify all documents that describe the activity to ensure they are updated accordingly.

The model also revealed areas where duplicate testing was taking place. This is an excellent example of using the model to identify activities within the process that can be modified almost immediately to provide savings in the effort and time to produce modifications to the software.

We feel that one of the most powerful benefits that the process model provides is in the area of metrics identification and collection. The model directly supports an activity-based cost (ABC) modelling approach. The model should contain more than one level of detail describing basic software maintenance activities. The SMO can use this model to determine what level of detail is desirable for metrics collection. Because each activity in the model (regardless of level of abstraction) identifies all organizations involved in the performance of the activity, the metric can be identified and an organization can be assigned the responsibility to collect the metric. If the model is used to generate the CM and QA procedures this information can be documented in the QA procedures.

Finally, even though this methodology was developed and used prior to the latest revision of the DoD 5000 series acquisition documents, we have found it to be in compliance with the spirit of these directives and instructions.

## 5. FUTURE DIRECTIONS

With the experiences that have been gained using this approach, we can now look to further refinement of our methodology and also toward future directions. The most logical next step is to incorporate the definition and use of metrics, both programmatic and technical, as part of the process model. This provides a foundation for further process improvements and also aids the practitioners in understanding when certain metrics should be collected. The incorporation of metrics also provides the basis for developing an activity-based cost model. ABC models can assist an organization in understanding the costs for each activity that it performs and therefore aid in process improvement and cost reduction.

The majority of this process modelling work was performed using the $IDEF_0$ methodology. However, a newer methodology in the IDEF family has emerged (with supporting tools) that could allow for the development of better process models. $IDEF_3$ can be distinguished from other process modelling methods because it facilitates the capture of the description of what a system actually does. The $IDEF_3$ method uses a knowledge acquisition strategy centred on the capture of descriptions of process flow (processes and their temporal, causal and logical relations) along with identification of objects that participate in these processes and the state transitions of those objects (Mayer, *et al.*, 1992). This method also allows for the capture of partial information where the complete description is not known or well understood. These features provide some distinct modelling advantages over the traditional $IDEF_0$ modelling methodology. In the future, consideration should be given to using the most appropriate modelling methodology to invoke process assessments and improvements in an organization.

## Acknowledgements

We acknowledge our thanks and appreciation for the helpful comments of the *Journal's* anonymous reviewers and encouragement of Dr Ned Chapin, the Editor.

## References

Bollinger, T. B. and McGowan, C. L. (1991). 'A critical look at software capability evaluation', *IEEE Software*, **8**(7), 25–41.

Humphrey, W. (1989) *Managing the Software Process*, Addison-Wesley, Reading, MA, 494pp.

Marca, D. A. and McGowan, C. L. (1993) *$IDEF_0$/SADT Business Process and Enterprise Modeling*. Eclectic Solutions, San Diego, CA, 392pp.

Mayer, R. J., Cullinane, T. P., de Witte, P. S., Knappenberger, W. B., Perakath, B. and Wells, M. S. (1992) *Information Integration for Concurrent Engineering $IDEF_3$ Process Description Capture Method Report*, AL-TR-1992-0057, KBSI, College Station, TX, 136pp.

McGowan, C. L. and Bohner, S. A. (1993) 'Model based process assessments', in *Proceedings of the 15th International Conference on Software Engineering*, Baltimore, MD, IEEE Computer Society Press, Los Alamitos CA, pp. 227–231.

Paulk, M. C., Curtis, B., Chrissis, M. B. and Webber, C. V. (1991) *Capability Maturity Model for Software*, CMU/SEI-91-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 64pp.

Schrank, M. J., Anderson, A. C., Bisignani, M. E. and Boyce Jr., G. W. (1995) 'Assessing the capabilities of military software maintenance organizations', in *14th Digital Avionics System*

 

*Conference Proceedings*, Cambridge, MA, IEEE Computer Society Press, Los Alamitos, CA, pp. 97–105.

**Authors' biographies:**

**Derick E. Masengale** is a Project Manager in the Electronic Data Systems (EDS) Government Services Business Unit located in Falls Church, Virginia. His current work involves developing and implementing the processes and procedures required to produce data warehouses in support of the executive information systems (EIS)/decision support systems (DSS) requirements of organizations. Derick has a B. S. degree in Computer Engineering from the University of South Florida, in Tampa, Florida. Email: dmasenga@med.osd.mil

**Michael J. Schrank** is a Senior Software Systems Engineer in the Software Engineer Center (SWEC) at The MITRE Corporation located in Reston, Virginia. His current work involves providing acquisition support and analysis for real-time embedded systems and distributed information systems; and assessing software products and processes during development and maintenance. Mike has a B.S. degree in Electrical Engineering from Purdue University; an M. S. degree in Engineering from Washington University in St. Louis, Missouri; and is currently working on an M. S. degree in Computer Science from George Mason University in Fairfax, Virginia. Email: mschrank@mitre.org

 